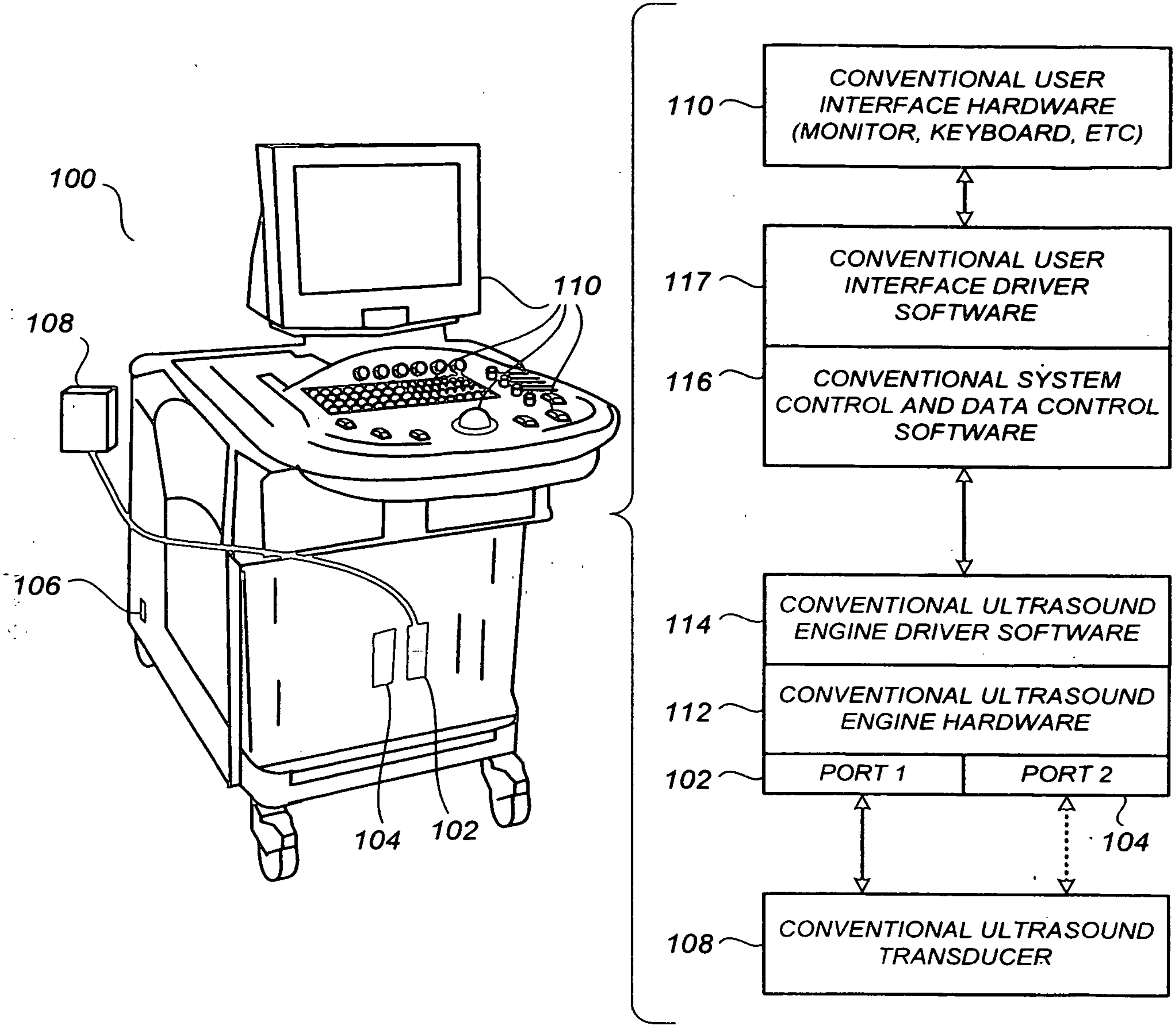
100
108
110
106
104
102
110
CONVENTIONAL USER INTERFACE HARDWARE (MONITOR, KEYBOARD, ETC)
117
CONVENTIONAL USER INTERFACE DRIVER SOFTWARE
116
CONVENTIONAL SYSTEM CONTROL AND DATA CONTROL SOFTWARE
114
CONVENTIONAL ULTRASOUND ENGINE DRIVER SOFTWARE
112
CONVENTIONAL ULTRASOUND ENGINE HARDWARE
102
PORT 1
PORT 2
104
108
CONVENTIONAL ULTRASOUND TRANSDUCER

PRIOR ART

FIG. 1

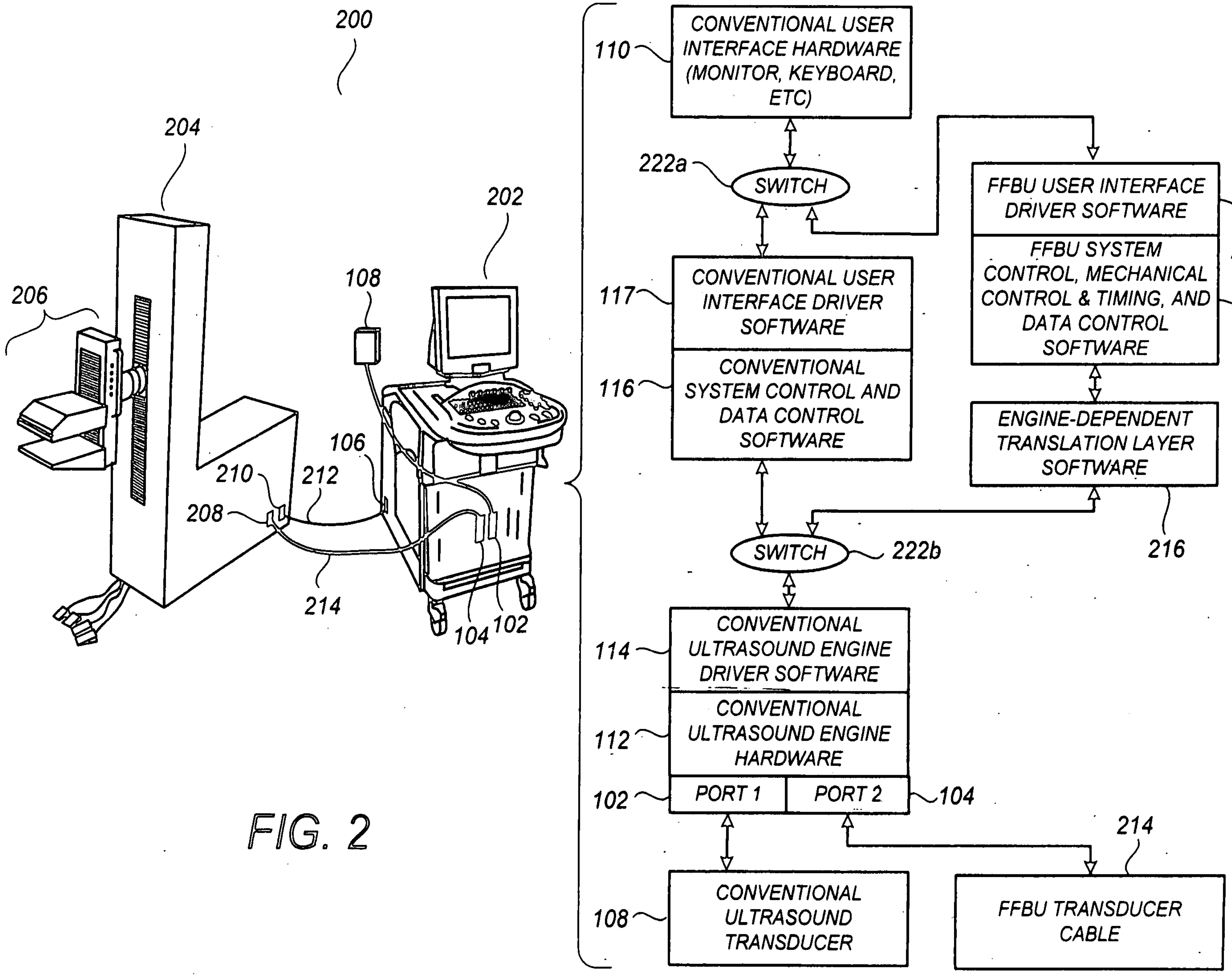
200
204
206
202
108
106
210
212
208
214
104
102
CONVENTIONAL USER INTERFACE HARDWARE (MONITOR, KEYBOARD, ETC)
110
222a
SWITCH
FFBU USER INTERFACE DRIVER SOFTWARE
220
FFBU SYSTEM CONTROL, MECHANICAL CONTROL & TIMING, AND DATA CONTROL SOFTWARE
218
117
CONVENTIONAL USER INTERFACE DRIVER SOFTWARE
116
CONVENTIONAL SYSTEM CONTROL AND DATA CONTROL SOFTWARE
ENGINE-DEPENDENT TRANSLATION LAYER SOFTWARE
216
SWITCH
222b
114
CONVENTIONAL ULTRASOUND ENGINE DRIVER SOFTWARE
112
CONVENTIONAL ULTRASOUND ENGINE HARDWARE
102
PORT 1
PORT 2
104
214
108
CONVENTIONAL ULTRASOUND TRANSDUCER
FFBU TRANSDUCER CABLE

FIG. 2

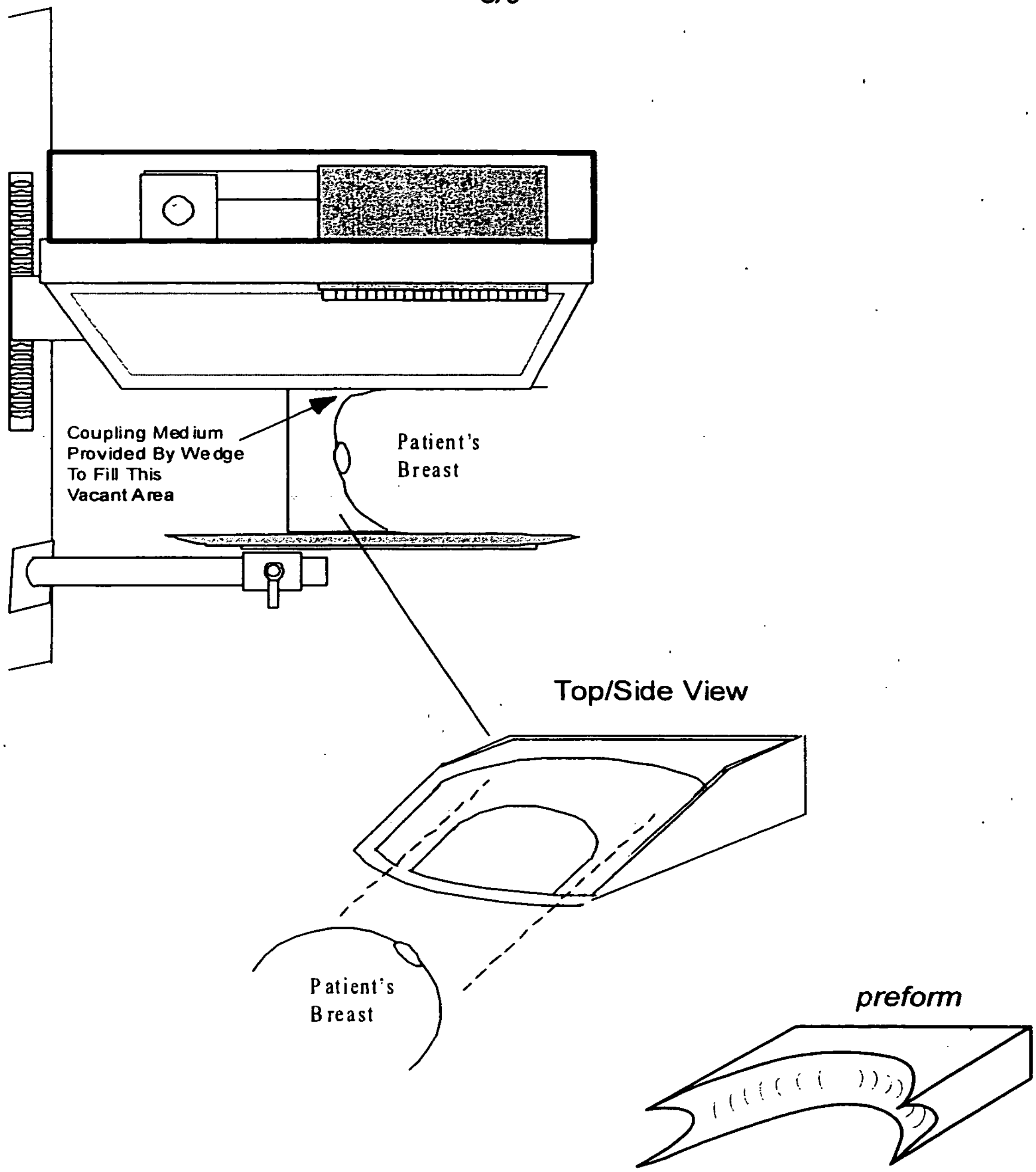
Coupling Medium
Provided By Wedge
To Fill This
Vacant Area
Patient's
Breast
Top/Side View
Patient's
Breast
preform

FIG. 3

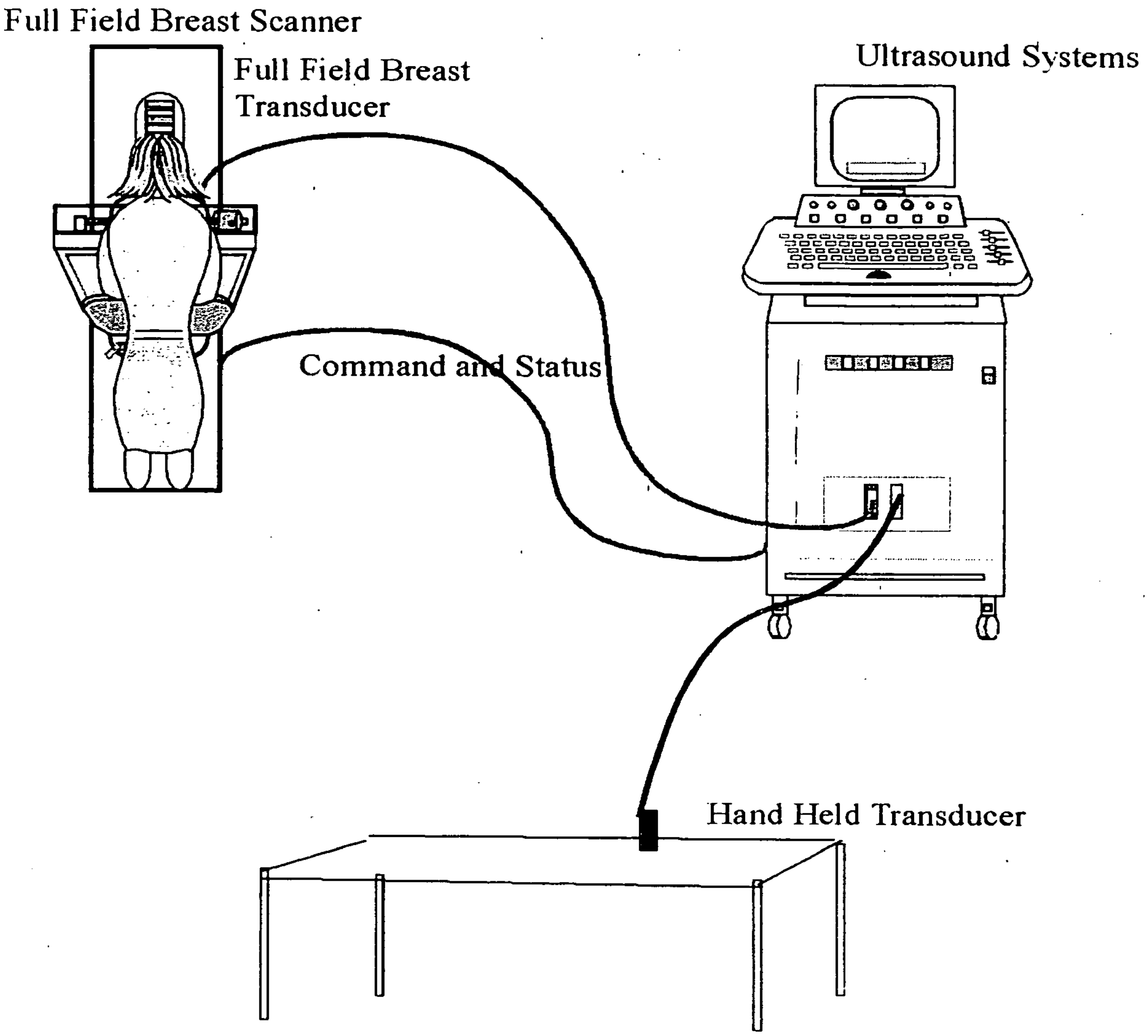
Full Field Breast Scanner
Full Field Breast Transducer
Ultrasound Systems
Command and Status
Hand Held Transducer

FIG. 4

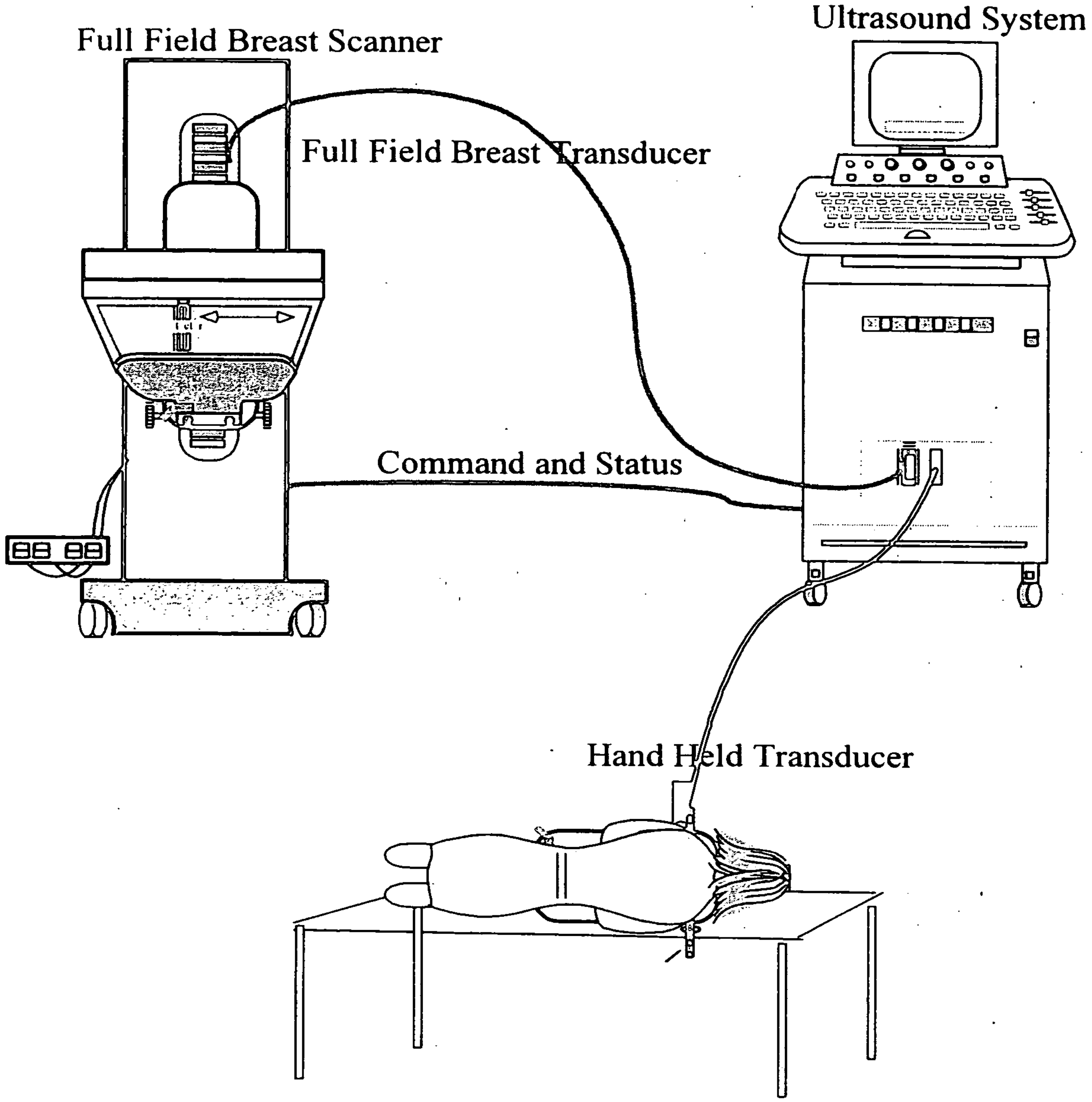
Full Field Breast Scanner
Ultrasound System
Full Field Breast Transducer
Command and Status
Hand Held Transducer

FIG. 5

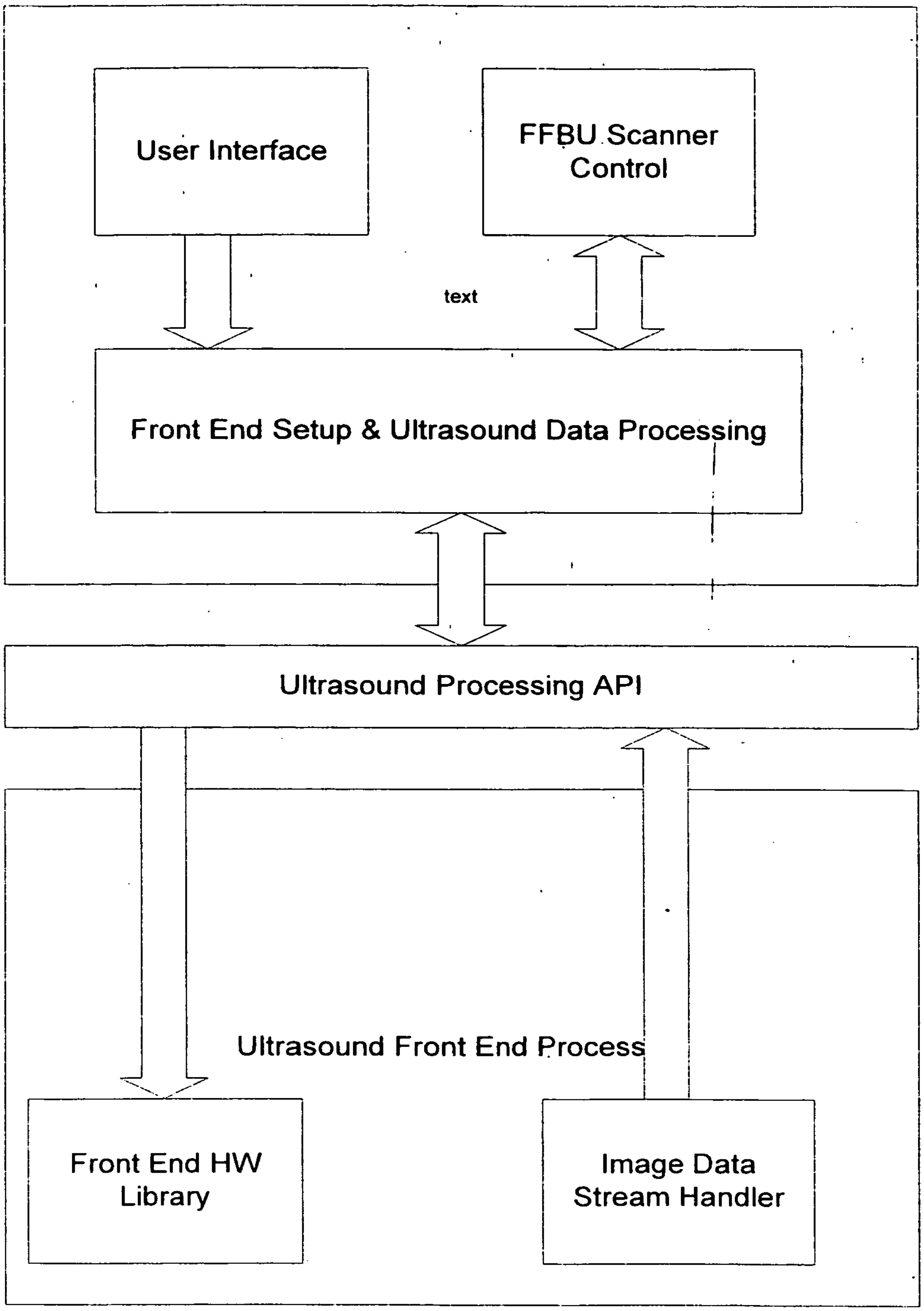
User Interface
FFBU Scanner Control
text
Front End Setup & Ultrasound Data Processing
Ultrasound Processing API
Ultrasound Front End Process
Front End HW Library
Image Data Stream Handler

FIG. 6

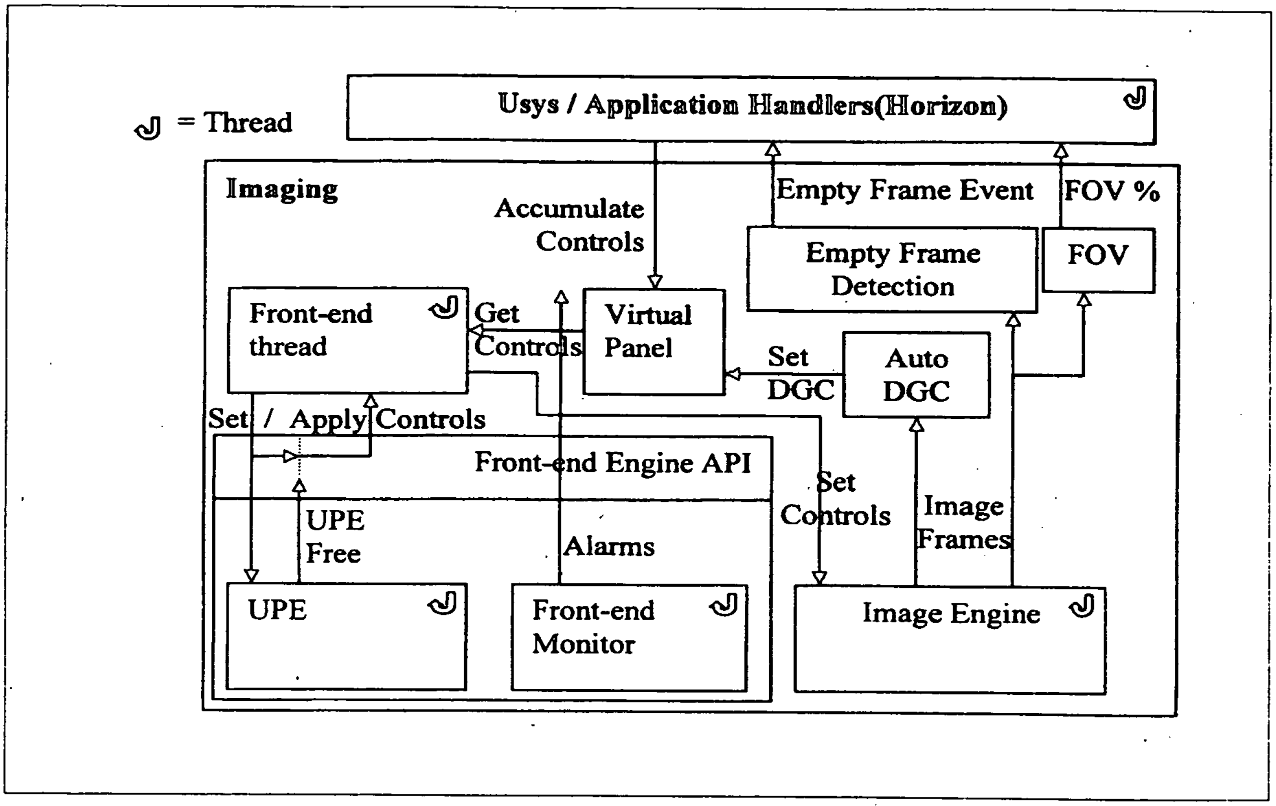
= Thread
Usys / Application Handlers(Horizon)
Imaging
Accumulate Controls
Empty Frame Event
FOV %
Empty Frame Detection
FOV
Front-end thread
Get Controls
Virtual Panel
Set DGC
Auto DGC
Set / Apply Controls
Front-end Engine API
Set Controls
Image Frames
UPE Free
Alarms
UPE
Front-end Monitor
Image Engine

FIG. 7

Type Definitions

```
enum EImagingModes { kModeB, kModeBColorFlow, kBModeColorPower,
                     ... };
enum EImagingControls { kDGC1, ..., kDGCn, kGainB, kDepth, kFOV,
                        kFOVMode, ... };
enum EConvertModes { kNearestNeighbor, kBilinearInterpolation, ... };
struct SProbeParameters { std::string sProbeName, int iNumElements, double
                          dElementPitchMM, ...};

struct SAlarmStatus { BOOL PowerStatus, int iCageTemperature, ... };
```

Imaging Control

```
Initialize();
            Initializes the imaging system, including loading the probe
            tables and establishing communication to the embedded system.
Terminate();
            Shuts down communication to the embedded system and powers
            it off.
vector< std::string> ReadProbeConnectorStatus();
            Retrieves a vector of length equal to the number of probe
            connectors. Each entry contains the name of the probe installed
            in that connector or "" if no probe is installed.
SelectProbeConnector( int iConnector );
            Enables the probe in the specified connector to acquire images
            and enables the probe table for that probe.
SProbeParameters GetProbeParameters( std::string sProbeName );
            Returns the probe parameters for the probe with the specified
            ProbeId.
int GetActiveProbeConnector( );
            Returns the connector index of the currently enabled probe.
vector< std::string> GetApplicationList( std::string sProbeName );
            Returns the list of the applications supported for the specified
            probe ID.
```

FIG. 8-1

Imaging Control (cont.)

vector< std::string> GetSubApplicationList(std::string sProbeName, std::string sApplicationName);

Returns the list of the sub applications supported for the specified probe ID and application.

SelectProbeApplication(std::string sApplicationName, std::string sSubApplicationName);

Selects the probe program associated with the specified program name for the currently selected probe.

SetImagingMode(EImagingModes);

Set the imaging mode to the specified mode.

EImagingModes GetImagingMode( );

Returns the currently active imaging mode.

SetImagingControl(EImagingControls eImagingControl, double dValue);

Sets the specified user imaging control to the specified value.

ApplyImagingControls( );

Applies the current set of imaging controls to the imaging HW. Blocks until the parameters are completely applied to the imaging HW.

double GetImagingControl(EImagingControls eImagingControl);

Returns the current setting of the specified imaging control.

const SDisplayParameters GetDisplayParameters( );

Read the current set of derived display parameters.

Freeze(BOOL bFreeze);

Immediately freezes or unfreezes imaging, retaining the current imaging mode. Unfreeze will automatically apply any controls that are still pending.

BOOL GetFreezeStatus( );

Returns the current freeze state.

SAlarmStatus GetAlarmStatus( );

Returns the current front end alarm status.

Attach(Aspect aNotificationAspect);

Attaches for notification on the specified aspect. Aspects include:
Probe Inserted Or Removed
Front End Alarm

Diagnostics / Engineering Tools

LoadProbeFile( );

Reloads and applies the probe table for the active probe using the currently selected probe program.

FIG. 8-2